Distributed Caching in Small Cell Networks

Kenza Hamidouche

Master research Internship

Research Internship Report

Distributed Caching in Small Cell Netwoks

Author:
Kenza Hamidouche

Supervisors:
Mérouane Debbah
Walid SAAD

Team: Alcatel-Lucent Chair on Flexible Radio

June, 2013

Abstract

The dense deployment of small cells in indoor and outdoor areas contributes mainly in increasing the capacity of cellular networks. On the other hand, the high number of deployed base stations coupled with the increasing growth of data traffic have prompted the apparition of base stations fitted with storage capacity to avoid network saturation. The storage devices are used as caching units to overcome the limited backhaul capacity in small cells networks (SCNs). Extending the concept of storage to SCNs, gives rise to many new challenges related to the specific characteristics of these networks such as the heterogeneity of the base stations. Formulating the caching problem while taking into account all these specific characteristics with the aim to satisfy the users expectations result in combinatorial optimization problems. However, classical optimization tools do not ensure the optimality of the provided solutions or often the proposed algorithms have an exponential complexity.

While most of the existing works are based on the classical optimization tools, in this thesis, we explore another approache to provide a practical solution for the caching problem. In particular, we focus on *matching theory* which is a game theoretic approach that provides mathematical tools to formulate, analyze and understand scenarios between sets of players. We model the caching problem as a one-to-one matching game between a set of files and a set of base stations and then, we propose an iterative extension of the *deferred acceptance algorithm* that finds a stable and optimal matching between the two sets. The experimental results show that the proposed algorithm reduces the backhaul load by 10-15 % compared to a random caching algorithm.

Contents

1 Introduction

During the last decade, mobile data traffic has grown exponentially, and is expected to continue increasing from 1.3 Exabytes per month by 2012 up to 10.8 Exabytes per month by 2016 [1]. Meanwhile, conventional wireless networks have already reached their capacity limits, especially during peak hours. Moreover, most of the existing solutions aiming to increase the data throughput through using more spectrum and improving spectral efficiency are not sufficient to keep up with traffic demand growth [3]. In order to increase the conventional network capacity and improve the users' Quality of Experience (QoE), the concept of *small cell networks* (SCNs) has recently emerged as a promising solution [2].

The SCNs' concept revolves around the very dense deployment of low-cost and low-power small base stations (SBSs) which provide a cost-effective way to boost the wireless capacity and offload traffic from the main macro-cellular networks. However, despite the advantages of the SCNs, the SBSs are connected to the network through a capacity-limited and possibly heterogeneous backhaul links which can reduce the potential gains of SCN deployments. To deal with the backhaul bottleneck, distributed caching in SCNs is considered as a promising solution [6], [7]. The basic idea of distributed caching is to duplicate and store the data in SBSs so as to serve users' requests locally, from the closest SBSs without using the backhaul links, whenever possible. However, storing data at the SCNs' level introduce several new challenges such as: which files should be cached, how files should be stored among the SBSs, what is the optimal caching strategy that would reduce the load of backhaul links and meet the users' requirements, among others. In most of the works that address the caching problem, the authors formulate the problem as combinatorial optimization problems, and then propose algorithms based on heuristics as solutions. These algorithms do not ensure the optimality of the solutions or often have exponential complexity. In this thesis, instead of using the classical optimization theory to answer to the addressed questions, we explore other approaches that provide practical solution for such problems. In particular, we focus on *matching theory* which is a game theoretic approach that provides mathematical tools to formulate, analyze, and understand scenarios between sets of players.

Matching games were firstly introduced to solve economics' problems such as the well-known *marriage problem* [12]. This problem consists of finding a matching between a set of men and a set of women while taking into account the preferences of each person over the set of the opposite sex. In our model, we consider two sets of players: a set $\mathcal{M}$ of SBSs and a set $\mathcal{C}$ of content provider servers (CPSs) that acts on behalf of a library of files. We aim using matching games to find an assignment of the files stored at the CPSs to the SBSs according to the popularity of files at each SBS as well as the users' expectation in terms of delivery time.

The remainder of this thesis is organized as follows. In Chapter 2, we provide a general introduction to SCNs and we discuss their limitations and challenges. The existing distributed caching techniques for SCNs are also presented in this chapter. Chapter 3 provides an introduction to matching theory. We first present the main

results in literature for each of the three classes of matching games: one-to-one, many-to-one and many-to-many matchings. Then, we provide an overview on three other matching classes: canonical matching, matching with externalities, and matching with dynamics. In Chapter 4, we model the storage problem in SCNs as a matching game and we propose an algorithm that aims to reduce the backhaul load. We also study its characteristics in this section. In Chapter 5, we evaluate the performances of the proposed algorithm compared to a baseline algorithm. Finally, we conclude in Chapter 6 with a brief summary and perspectives concerning the extensions of this work.

2 A general Overview on Small Cell Networks

2.1 Introduction

Recently, the wireless traffic has grown exponentially due to the proliferation of smartphones and tablets. Most of the traffic load is generated by multimedia content that require a high Quality of Service (QoS). Consequently, novel approaches for modeling, analyzing, and designing wireless networks are needed in order to maintain tolerable delays and efficiently exploit the scarce radio resources. Indeed, conventional networks have already met their capacity limits since the performance of the previously used solutions to enhance the networks capacity, meet with the ergodic capacity limit [3]. To overcome this issue and maintain seamless, high QoS wireless connectivity, SBSs were integrated to the conventional wireless networks resulting in the concept of small cell networks.

SCNs can be seen as a promising solution to increase network capacity with low energy consumption. In these networks, the integrated SBSs carry all the data traffic, while the typical macro base stations ensure the area coverage. Despite the advantages of the SCNs, they have an important issue since SBSs are expected to connect to the core network via a capacity-limited backhaul links. Indeed, the solution of using fiber-optic to connect the SBSs to the network is expensive. To deal with these backhaul-level limitations and reduce the associated congestion problems in the network, the distributed storage (distributed caching) over SBSs in SCNs is a new promising solution proposed in [6], [7]. Nevertheless, several metrics should be considered in the proposed distributed caching approaches. These metrics are related to both the specific characteristics of the networks and users' requirements. In particular, the most accessed data by users (users' preferences) and the mobility patterns of users need to be considered to define efficient caching strategies.

In this chapter, we first give a general introduction to the concept of SCNs as well as to the benefits from their deployment. Then, we discuss the main challenges that face the deployment and development of small cells. Furthermore, we address the problem of distributed storage in SCNs and we present the works investigating distributed storage solutions in SCNs.

2.2 Definitions and Generalities

In this section we define the concept of SCNs and we discuss their limits.

- **Small cells:** This term includes all low-powered radio access base stations that operate in a range spectrum of 10 meters to several hundred meters, in contrast to the macrocells that operate in a range up to several tens of kilometers. Small cells include: femtocells (10 m), picocells (100 m) and microcells (1km-3km) [3], [5]. Grouping these different small cells with the typical macrocells give rise to a SCN.

- **Small Cell Networks:** SCNs are heterogeneous networks that comprise all forms of cells. They are based on a very dense deployment of small cells in order to provide superior cellular coverage in residential, enterprise, or hot spot outdoor environments (see Figure 1).

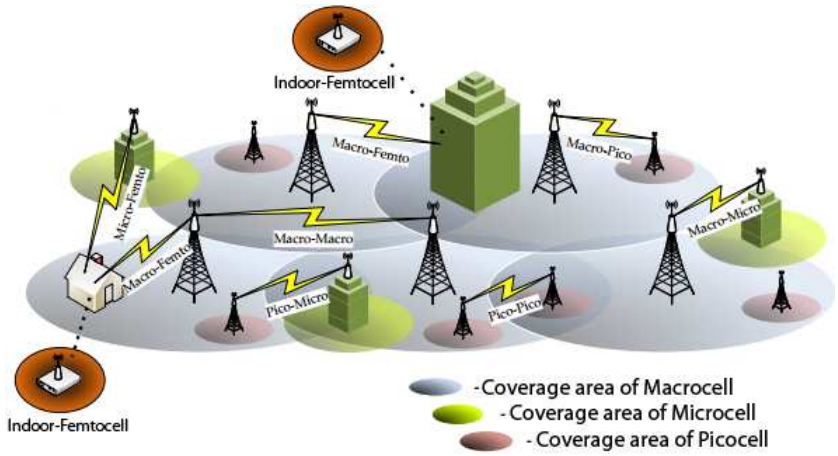

Figure 1: A typical small cell network deployment.

In essence, the small cell technology was introduced as an ecological and economical solution to overcome the imminent wireless capacity crunch. Indeed, the impressive increase of sophisticated mobile devices (smartphone, laptop and tablet), coupled with the proliferation of social networks mainly contribute to the exponential growth of data traffic leading wireless networks to their capacity limits. The main benefits of including low-power, low-cost and self-organizing base stations to the traditional mobile networks are the following [3], [4]:

- Dense deployment of small cells closer to the users can reduce the number of devices sharing the bandwidth of a single cell. This provides a higher data-rate for each served user and hence improves the capacity and overall performance of the radio access network.

- The low power transmission in small cells compared to macro cells reduces the network energy consumption, if well designed. Furthermore, the deployment of self-powered small cells and the integration of advanced energy saving techniques can potentially enhance the network performance in terms of energy consumption.
- SCNs do not require any specific infrastructure since they share the existing point-to-point wireless and/or wired backhaul and they might be installed in the users' homes, street equipments, and enterprise buildings. Moreover, the unplanned deployment of small cells reduces the need of costly network planners and hence the operational expenses.

Despite the promising potential of SCNs, they present a number of limitations we need to consider when proposing solutions that aim to increase their capacity and performance such as reliability, throughput, etc.

2.3 Challenges Facing Small Cell Networks

This section discusses several challenges that face the development and the deployment of small cells [4]. The most pertinent challenges are:

- **Self-organization:** Since some small cells are user-deployed, they require self-configuration, self-optimization and self-healing capabilities. They need to be automatically configured before entering the operational state, optimize their parameters to improve coverage, reduce energy consumption and interference, and perform recovery from node failures. The insurance of the self-organization in SCNs is challenging due to the diversity of coexisting cells and their different characteristics which increase the number of parameters that need to be considered.
- **Handover:** The limited coverage range of small cells and the frequent mobility of users increase the call to the handover process. This one consists in transferring the users at the border of adjacent cells to less congested cell in order to improve the QoS. However, the existing handover mechanisms increase the system load due to the large number of cells. Hence, more appropriate procedures are required.
- **Interference management:** The unplanned user-deployment and the large number of BSs cause interference problem in SCNs, which arise from macro cells as well as from neighbor small cells. The interference problem might significantly deteriorate the system performance, hence, interference management techniques are needed to reduce interferences.
- **Backhaul management:** The backhaul infrastructure connect small cells to the core network. It presents several issues related to the lacks of security, QoS and the limited bandwidth.
  - Security: The communications are performed over uncontrolled backhaul infrastructure. Thus, secured mechanisms are required to ensure the confidentiality and the integrity of exchanged data.

  - QoS: No packet loss and minimal delay should be guaranteed to ensure a high QoS with unreliable links. For this, small cells may allow a handover to the macrocell when it is necessary.
  - Bandwidth limitation: The backhaul capacity is limited since using fiber-connections, as a solution, is expressive. To avoid creating a traffic bottleneck, one novel approach based on distributed storage was proposed in [6].

- **Predictive scheduling:** Many recent works focus on the prediction of users' behaviors. Actually, a number of predictors reach an average prediction accuracy of more than 60% [10]-65% [11]. The attractiveness of these results makes the prediction an important parameter that might allow the implementation of efficient algorithm for resource allocation in SCNs, given predictable behaviors of frequent users. Thus, new techniques that exploit the periodic behavior of users, the accessed content in a specific places and times, could be developed to improve users' QoE. This might include models for caching, pre-allocating resources over time and joint backhaul caching and radio resource allocation.

The limited backhaul capacity issue is the subject of this internship. Thereafter, we introduce the distributed-storage based solutions to deal with this problem.

2.4 Distributed Caching in Small Cell Networks

The main idea behind the introduction of distributed caching into SCNs is bringing data closer to the users such as caching replace the backhaul communication [6]. For this purpose, authors in [6] proposed to equip femtocells with large caching capacity and integrate to the network, fixed wireless storage nodes called *helpers*. These caching nodes store the most popular files and handle the mobile users requests locally. In the case of unavailability of data in their cache, the macro BSs serve the users.

This idea was extended in [7] by considering also the users as caching nodes. The data traffic between helpers and users is called device to device (D2D) communication. The authors described the architecture of a network as a set of square cells such as one macrocell serve N users in a given square. Each cell is then divided into smaller squares with equal size named, *clusters*. The assignment of data is performed centrally by a macrocell BS which offer an efficient assignment of data to the mobile users. Another proposed solution consists in randomly allocating data according to a given probability density function.

2.4.1 Related Works

The main goal of the proposed approach in [6] is to find an optimal allocation that minimizes the expected delay for data recovery. It is assumed that users may recover the data either from a central node (MC) or locally from a fixed number of storage nodes (SBSs). Indeed, once the optimal allocation is determined, the central node disperses the data according to their given popularity over the storage nodes. These nodes have a

limited storage capacity and the links between the nodes and the collector have a finite transmission rate.

Two cases were considered: uncoded and coded distributed storage. In the first version, the data are stored directly in the storage nodes. The authors showed that defining the optimal allocation is *NP-complete*. To solve this problem, they reformulate the minimization of the average delay of all collectors as a maximization of a submodular function [9] and showed that the storage capacity constraint forms a matroid constraint [9]. Thereby, they offer a greedy algorithm where at each step an assignment of data i to a specific storage node j is selected.

In the second version, the data are encoded and then stored over storage nodes. The average delay recovery is computed by taking into account that the collector needs to contact j storage nodes to recover the original object and $j-1$ nodes are not enough. Using this formulation, the obtained objective function which is convex and the constraint of the upper bound storage capacity is linear. Thus, the problem can be solved using standard convex optimization [8] as it can be turned into a linear program by introducing new variables.

2.4.2 Discussion

When data are distributed over the nodes of the network, the allocation process should take into account the dense deployment of small BSs. The existing allocation solutions propose to spread the data according to the objects popularity. Hence, a popular object can be stored in many nodes that are located in the same area which leads to waste of storage space. Moreover, the assignment of data considers only the global popularity of content, i.e., the proportion of users that would request each data. However, the popularity of data can differ from a SBS to another SBS depending on the preferences of users that would connect to each SBS. On another hand, the authors in [8] formulated the optimization problem in order to reduce the recovery delay, whereas they do not take into account the mobility of users. In fact, the expected performance such as the recovery delay is not static since it changes with topology changes over time. The proposed algorithm to assign data is a centralized algorithm, executed in a central application server. However, a distributed algorithm could be more appropriate in such a system in which SBSs are owned by different users or operators and data delivered by several content providers. Thus, the information about the stored data in each SBS might be private and for secrecy reasons the information should be kept local.

2.5 Conclusion

The increasing demand of data prompted the apparition of SCNs as an ecological and economical solution to provide a better coverage and improve the capacity performance of cellular networks. However, SCNs present some limitations and the most significant is the limited backhaul capacity. To deal with this problem, distributed caching over SCNs was proposed as a solution. The main idea behind distributed caching is to bring data as close as possible to users by storing them and introducing redundancies in the

small cells. This concept would allow small cells to serve users requests locally and thus reducing the backhaul links load.

In order to take a maximal benefit from the caching process in SCNs, optimal caching strategies need to be developed. In general, the caching problem is formulated as optimization problems which are mostly combinatorial or have exponential complexity. In the following chapter we present *game theory* as an alternative solution to solve these problems. More specifically, we will introduce *matching games* which were mainly proposed to provide mathematical tools for modeling many economical systems and afterward, have been extensively used in literature to solve allocation problems in telecommunication systems.

3 Introduction to Matching Theory

3.1 Introduction

In the previous chapter we showed that self-organization can be seen as the most important feature in small cell networks which allows terminals and base stations to interact and self-adapt intelligently without any external or central control. The importance of self-organization motivates researchers to exploit the intelligence of network equipments by addressing the wireless communications' problems in a distributed manner. For instance, resources time, spectrum, and space are limited, while the demand for mobile communication and services have grown exponentially, increasing the need of new techniques to manage and allocate these resources efficiently. Therefore, resource allocation problems such as power allocation, spectrum allocation, user grouping, user selection, resource assignment, and beamforming optimization, are widely studied, and are often formulated as optimization problems. However, most of the resource allocation problems are combinatorial. The exponential complexity of these problems prompts the introduction of new centralized and distributed mechanisms, different from the conventional optimization theory. Game theory is one of approaches extensively used for modeling. It was mainly introduced to explain and predict the behavior of complex real world economic systems. The theory was then developed widely and applied to many fields such as biology, computer science, philosophy and telecommunication. Formally, game theory is a set of suitable mathematical tools that provides a language to formulate, analyze and understand scenarios between competing or cooperating players.

In this work, we focus on the game-theoretic matching problem which has been adopted in many real-world systems such as the assignment of interns to hospitals in United States. Especially, we introduce the two-sided matching games which refer to the markets of bilateral nature. In such a market, players belong to only one of two disjoint sets. For instance, in the interns and hospitals market, if an intern decides to not do his/her internship and quit the market, he/she cannot join the other side of the market and become a hospital. The members of the two sets are called and they want to be matched. Depending on the number of partners to which an agent of each side, wants to be matched to, the two-sided matching games can be divided into three categories: one-to-one, many-to-one and many-to-many matchings. In one-to-one matching, each agent is matched to one agent of the opposite set (e.g., marriage market), while in the many-to-one matching, in one side, agents are matched to one agent and in the other side, agents are matched to a group of agents (e.g., colleges admissions market). Many-to-many matching is a generalized model where each agent of the two sides can be matched to many agents of the other side (e.g., firms and workers). Matching games are also classified into three other classes: canonical matching games, matching with externalities and matching with dynamics.

In the remainder of this chapter, we first detail the three classes of matching games: one-to-one, many-to-one and many-to-many matching games, by giving the formal description of the corresponding real-world markets and the main results in literature. We then provide an overview of the three last classes: canonical games, matching with ex-

ternalities and matching with dynamics. Finally, we conclude this chapter with a brief summary.

3.2 One-to-One Matching Model

One of the most popular matching problems is the stable marriage problem, where a set of women and a set of men decide to whom to get married. The marriage model is a two-sided one-to-one matching problem where each agent aims to be matched to one agent of the opposite set, according to his/her preferences. In fact, every agent has a preference list over the set of the opposite sex which could be related to face feature, eyes color, hair length, etc. The agent's preferences are called *strict* when the agent is not indifferent between any two partners of the other side, or between being matched to a partner and being unmatched.

3.2.1 Centralized Stable Matching for Marriage Problem

The formal description of the marriage model is given as follows. We consider an instance $\mathcal{I}$ of the stable marriage problem involving N men and M women. The two finite disjoint sets of men and women are denoted $\mathcal{M} = \{m_1, m_2, \ldots, m_N\}$ and $\mathcal{W} = \{w_1, w_2, \ldots, w_M\}$, respectively. Each agent ranks agents of the opposite set to form his/her preferences list of the form $p_{m_1} = w_1, w_2, ..., \emptyset$, meaning that m_1 prefers w_1 to w_2 and so on, until he/she prefers to remain single. The preference relation of an agent k over the set of his/her potential partners is denoted $\prec_k$. $w_1 \prec_{m_1} w_2$ means that man m_1 prefers to be matched to woman w_1 than to be matched to woman w_2. When an agent k prefers to remain single than to be matched to an agent l, l is said to be *unacceptable* to k, and this preference is denoted $\emptyset \prec_k l$. The outcome of the marriage market is a matching μ defined as follows [12]:

Definition 3.1. *A matching μ is a function from the set $\mathcal{M} \times \mathcal{W}$ onto the set $\mathcal{M} \times \mathcal{W}$ such that:*

1. $|\mu(k)| = 1$ *for every* $k \in \mathcal{W} \cup \mathcal{M}$;
2. $\mu(m) \in \mathcal{W} \cup \emptyset$ *and* $\mu(w) \in \mathcal{M} \cup \emptyset$;
3. $w = \mu(m)$ *if and only if* $m = \mu(w)$.

So $\mu(m) = w$ denotes that man m is a partner of woman w under the matching μ. Similarly, $\mu(w) = m$ denotes that woman w is a partner of man m under the matching μ. In a marriage market, the main goal is to form a matching where agents do not have an incentive to break up and form new marriages. Such a matching is called *stable matching*. For this purpose, many criteria should be considered such as the individual rationality and blocking partners [12].

Definition 3.2. *A matching μ is individually rational to all agents, if and only if there does not exist an agent k who prefers being single to being matched with $\mu(k)$, i.e., $\mu(k) \prec_k \emptyset$.*

Definition 3.3. *A matching μ is blocked by a pair of agents (m, w) if they each prefer each other to the partner they receive at μ, i.e., $\mu(m) \prec_m w$ and $\mu(w) \prec_w m$.*

Thus, a stable matching could be defined as follows:

Definition 3.4. *A matching μ is stable if and only if it is individual rational, and is not blocked by any pair of agents.*

In[12], the authors showed that:

Theorem 3.1. *A stable matching exists for every marriage market.*

They proved the existence of a stable matching by proposing a *deferred acceptance algorithm* (*Gale-Shapley algorithm*), which always finds a stable set of marriages. The deferred acceptance algorithm works as follows:

- **Step 1:** Each agent defines his/her strict preference list over the agents of the opposite side. When an agent is indifferent between two partners, an order is defined randomly or based on a specific metric such as the alphabetic order;
- **Step 2:**
  1. Each man proposes to his favorite partner if he has any acceptable choices;
  2. Each woman keeps the most preferred proposal among the acceptable ones if any, and rejects the others;
- **Step 3:**
  1. Each rejected man proposes to his favorite woman who has not rejected him yet;
  2. Each woman chooses her favorite partner from the group of the new proposers and the proposer kept at the previous step;
- **Step 4:** Redo Step 3 until no proposal could be made, i.e. each man was either accepted by a woman or rejected by all the women of his preference list.

The optimality of the resulting matching μ of the deferred acceptance algorithm is also an important parameter. A matching μ is called $\mathcal{M}$-optimal if every man likes it at least as well as any other stable matching. Similarly, a matching μ is $\mathcal{W}$-optimal if every woman likes it at least as well as any other stable matching. The authors showed that such optimal stable matchings do exist [12].

Theorem 3.2. *The resulting matching of the deferred acceptance algorithm with men proposing is M-optimal, whereas the resulting matching of the deferred acceptance algorithm with women proposing is W-optimal.*

The optimality of the stable matching was proved by showing that, in the deferred acceptance algorithm, no man is ever rejected by a woman who could be matched to him at a stable matching.

One of the most important concepts in matching theory is the *core* of the game. In the marriage market, rules of the game consist in the agreement of the two sets agents on their matching. The rules of the game coupled with the preferences of players induce a relation on the outcomes, called the *domination* relation, where a matching μ dominates a matching μ' if there is a coalition (i.e., subset) S whose members have both the incentive and the means to replace μ' with μ [**?**].

Definition 3.5. *The core of the game is the set of undominated outcomes.*

The difference between the core of a game and the set of a stable matching relies on the fact that an outcome fails to be in the core if it is blocked by any coalition of agents, whereas it fails to be in a stable matching only if it is blocked by some individual agent or by some pair of agents consisting of a man and a woman.

In the marriage market, even when coalitions other than singletons and pairs are considered, the properties of the market remain correct [**?**].

Theorem 3.3. *The core of the marriage market equals the set of stable matchings.*

The proposed deferred acceptance algorithm is a centralized algorithm, i.e., the algorithm is executed in a central agent which should know all the players preference lists as well as their partners from the resulting matching. However, the players may want to do not disclose their preference list and keep secret. For this purpose, a distributed version of deferred acceptance algorithm was proposed in [21].

3.2.2 Distributed Stable Matching for Marriage Problem

The distributed version of Gale-Shapley algorithm, proposed in [21], consists of two procedures for men and women. The men and women procedures are executed asynchronously, in each man and each woman agent, respectively. The following messages are exchanged between men and women:

- **Propose:** Men send this message to women to propose engagement;
- **Accept:** Women send this message to men after receiving a proposal message to notify acceptance;
- **Delete:** Women send this message to men to notify that they are not available for the proposing men; this occurs either (i) proposing man an engagement to a woman but she has a better partner or (ii) a woman acceptant an engagement with other man more preferred than the proposing one;
- **Stop:** This is a special message to notify that execution must end; it is sent by an special agent after detecting quiescence.

At the initialization step, all the men and women do not have partners. On one hand, each man executes the following steps:

- **Step 1:** If a man is free and his preference list is not empty, he sends a proposition to the first woman in his preference list;
- **Step 2:** Wait for a response message;
- **Step 3:** If the response is a *accept* message, the man confirms the engagement; otherwise (i.e., the response is a *delete* message), the man deletes the sender from his preference list and he becomes free.
- **Step 4:** Redo the Steps 1-3 until receiving a *stop* message.

On the other hand, each woman executes the following steps:

- **Step 1:** When a woman receives a *propose* message from a man, she rejects him if he is not in her preference list. Otherwise, she accepts him by sending a message *accept*;
- **Step 2:** The woman sends a message *reject* to all the men who appear after the accepted one in the previous step, in her preference list. Also, she removes all these men from her preference list;
- **Step3:** Redo Step 1-2 until receiving a *stop* message.

3.3 Many-to-One Matching Model

3.3.1 The College Admissions Model with *Responsive* Preferences

Another well-known two-sided matching problem is *college admissions market* which involves a set of N colleges and a set of M students. The question that arose in this market is which and how many students will be admitted at each college, knowing that each one can offer admissions to only q applicants. There are many other matching problems that fit with the many-to-one problem description, where agents are individual on one side of the market and on the other side agents are institutions, such as: firms and workers, hospitals and medical students [12].

Formally, in the college admissions game, the two finite and disjoint sets of colleges and students are denoted $C = \{c_1, c_2, ..., c_N\}$ and $S = \{s_1, s_2, ..., s_M\}$, respectively. As in the marriage market, each agent has a preference list over the agents of the opposite set. In this model, each college c_i can accept q_i ($q_i \geq 1$) students, known as quotas $Q = \{q_1, q_2, ..., q_N\}$, whereas in the the marriage market all the agents accept only one partner. The marriage game could be seen as a special case of the college admissions game where colleges' quotas equal one ($q_i = 1, \forall i = 1, ..., N$). The quota of each college is represented by a waiting list of students. We note that no student is assigned to more than one college and no college is assigned more than its quota of students [**?**].

Definition 3.6. *A matching μ is a function from the set $\mathcal{C} \cup \mathcal{S}$ into the set of unordered families of the elements of $\mathcal{C} \cup \mathcal{S}$ such that:*

1. $|\mu(s)| = 1$ *for every* s, *and* $\mu(s) = \emptyset$ *if* $\mu(s) \notin \mathcal{C}$;
2. $|\mu(c)| = q_c$ *for every college* c, *and if the number of students in* $\mu(c)$, *say* r, *is less than* q_c, *then* $\mu(c)$ *contains* $q_c - r$ *copies of* $\emptyset$;
3. $\mu(s) = c$ *if and only if* $s \in \mu(c)$.

So $\mu(s) = c$ denotes that student s is enrolled at college c under the matching μ, and $\mu(c) = \{s_i, s_j, \emptyset\}$ denotes that college c, with quota $q_c = 3$, enrolls students s_i and s_j and has one position unfilled.

An extension of the marriage model was examined in order to preserve its properties in the admissions problem. It was shown that the Theorems 3.1 and 3.2 can be extended to the college admissions problem. However, Theorem 3.3 could not be extended because of the incomplete information related to the preferences of colleges over the outcomes. In fact, to state this theorem, each college should specify its preferences over sets of students, while in the description of this model, preferences of students and colleges are over individuals. Thus, colleges with quotas greater than one are not able to compare groups of students. To define a complete model and allow colleges to compare outcomes of matchings, the authors in [17] introduced the concept of *responsive* preferences.

Definition 3.7. *The preferences are called responsive if: for all set S with $|S| < q_i$ and any students s_i and $s_j \notin S$, c_i prefers $S \cup s_i$ to $S \cup s_j$ if and only if s_i is preferred to s_j under c_i's preferences over individual students, and prefers $S \cup s_i$ to S if s_i is acceptable to c_i.*

It should be noted that a college could be indifferent between two outcomes. For instance, we suppose a college c with a quota q_c=2 and a preference list $P(c) = s_1, s_2, s_4, s_3, \emptyset$, over a set of students $C = \{s_1, s_2, s_3, s_4\}$. When preferences are responsive, college c would prefer to admit students $\{s_1, s_2\}$ than to admit students $\{s_1, s_3\}$, because student s_2 is more preferred than student s_3. However, college c is indifferent between admitting the two sets of students $\{s_1, s_4\}$ and $\{s_2, s_3\}$.
The authors in [17], showed that even when the specification of the model are complete, Theorem 3.3 will be false as long as the preferences of colleges for sets of students are related to their preferences over individual students.

In [12], because of the responsiveness of preferences, the deferred acceptance algorithm of the marriage problem was extended to the college admissions problem. The authors in [16], gave a formal proof of the equivalence of the two problems by replacing each college c_i by q_i copies of c_i denoted $c_{11}, c_{12}, ..., c_{1q_i}$. Each of these c_{ij} has preferences identical with those of c_i but with quota 1. Each student who has c_i in his preference list replaces c_i by $c_{11}, c_{12}, ..., c_{1q_i}$, in that order of preference. Thus, they formulated the original problem as a one-to-one matching. The deferred acceptance algorithm for the marriage problem was modified and adapted to the college admissions problem [12]. The algorithm is given as follows:

- **Step 1:** All the students apply to the college of their first choice;
- **Step 2:** Each college places on its waiting lists the q_i most preferred students, or all the applicants if they are less than q_i, and rejects the rest;
- **Step 3:** Rejected applicants apply again to their following choice;
- **Step 4:** Each college selects the top q_i from among the new applicants and those on its waiting list, pUEs these on its waiting list and rejects the rest;
- **Step5:** Redo Steps 3-4 until every applicant is either on a waiting list or has been rejected by every college.

The referred stability in this model is the same stability defined in the one-to-one matching problem, where the essential coalitions that cause instabilities are pairs of agents and individuals. This concept of stability is called *pairwise* stability. In many-to-one matching, coalitions consisting of multiple colleges and students need to be considered. A coalition A is said to be a blocking coalition for a matching μ, if students and colleges in A, by matching among themselves could all get an assignment preferable to μ [13]. Formally saying, a matching μ is blocked by a coalition A, if there exists another matching μ', such that for all students s, and for all colleges c in A,

1. $\mu'(s) \in A$, i.e., every student in A who is matched by μ' is matched to a college in A;
2. $\mu(s) \prec_s \mu'(s)$, i.e., every student in A prefers his new match to his old one;
3. $k \in \mu'(c)$ implies $k \in \mu'(c) \cup A$, i.e., every college in A is matched at μ' to new students only from A, although it may continue to be matched to its old students from $\mu'(c)$;
4. $\mu(c) \prec_c \mu'(c)$, i.e., every college in A prefers its new set of students to its old one.

Now, we define another kind of stability that consider all the coalitions [13].

Definition 3.8. *A group stable matching is one that is not blocked by any coalition.*

When preferences are responsive the two stability concepts, group stability and pairwise stability, are equivalent [13]. Hence, the set of group stable matchings is always non-empty.

3.3.2 The Firms-Workers Model with *Substitutable* Preferences

In the previous section, where preferences of colleges over the set of students are responsive, we saw that the theorem 3.1 could be extended to the college admissions problem. In other words, the set of stable matchings for the college admissions model with responsive preferences is always non-empty. In this section, we consider a weaker condition than responsiveness that preserve the non-emptiness of the set of stable matchings as well as

many other properties. These results are due to the fact that the set of stable matchings continues to be non-empty as long as colleges' preferences over groups of students are such that the colleges regard individual students more as substitutes than as complements. This means that a college will continue to want to accept a student s_i even if some of the other students become unavailable. In fact, this condition make the deferred acceptance algorithm operate as it did when the preferences are responsive.

To well describe the many-to-one matching with substitutable preferences, we consider the labor market firms and workers. The sets of firms and workers are denoted $\mathcal{F} = \{f_1, f_2, ..., f_N\}$ and $\mathcal{W} = \{w_1, w_2, ..., w_M\}$, respectively. Workers have preferences over individual firms, and firms have preferences over subsets of workers. The worker preference list is represented by a list of acceptable firms $P(w) = f_i, f_j, f_k, \emptyset$, and the firm preferences by a list of acceptable workers $P^{\#}(f) = S_1, S_2, ..., S_k, \emptyset$, where each S_i is a subset of the set of workers $\mathcal{W}$. We denote the preferences of all the agents $P = (P^{\#}(f_1), P^{\#}(f_2), ..., P^{\#}(f_N), P(w_1), P(w_2), ..., P(w_M))$.

When firms are faced to a set S of workers, each firm f defines its most preferred subset of S it would hire, which could be an empty set or equal to the set S. This set represents the firm f's choice from S, denoted $S' = Ch_f(S)$. Formally, the set $S' \subset S$ is defined as follows:

- $\forall S'' \subseteq S$, $S'' \preceq_w S'$.

Now, we can define the *substitutability* property of preferences.

Definition 3.9. *A firm f's preferences over sets of workers has the property of substitutability if, for any set S that contains workers w and w', if w is in $Ch_f(S)$ then w is in $Ch_f(S - w')$.*

According to this definition, we can see that responsiveness includes the substitutability property. In fact, the choice set from any set of students of a college with quota q is either the q most preferred acceptable students in the set, or all the acceptable students in the set. In this model, a matching μ is blocked by a firm f if $\mu(f) \neq Ch_f(\mu(f))$, i.e., the firm f would prefer to hire all the assigned workers by the matching μ. A matching μ might be blocked by an individual firm f without being individually irrational, since it might still be that $\emptyset \preceq_f \mu(f)$. Similarly, a matching μ is blocked by a worker-firm pair (w, f) if w and f are not matched at μ but would both prefer if f hired w; that is, if $\mu(w) \neq f$ and if $\mu(w) \prec_w f$ and $w \in Ch_f(\mu(f) \cup w)$. If the firms have responsive preferences, this definition is equivalent to the used one for college admissions problem.

Definition 3.10. *A matching μ is stable if it is not blocked by any individual agent or any worker-firm pair.*

Theorem 3.4. *When firms have substitutable preferences, the set of stable matching is always nonempty.*

This theorem was proved by modifying the deferred acceptance algorithm with firms proposing [13]. The modified version works as follows:

- **Step 1:** Each firm proposes to its most preferred set of workers;
- **Step 2:** Each worker rejects all but most preferred acceptable firm that proposes to him;
- **Step 3:** Each firm proposes to its most preferred set of workers including the workers whom it previously proposed to and who have not yet rejected it without including the works who have previously rejected him;
- **Step 4:** Each worker rejects all but the most preferred acceptable firm that has proposed so far;
- **Step 5:** Redo Steps 3-4 until there are no rejections.

Also, the authors showed that:

Theorem 3.5. *When firms have substitutable preferences and preferences are strict, the deferred acceptance algorithm with firms proposing produces a firm-optimal stable matching. Also, the deferred acceptance algorithm with workers proposing produces a worker-optimal stable matching.*

3.4 Many-to-Many Matching Model

The many-to-many matching problem is a generalized model of the two previous matching models: one-to-one and many-to-one matchings. To describe this model, we extend the labor market of firms and workers studied in many-to-one matching with substitutable preferences. Similarly, we suppose two sets of agents, firms and workers denoted $\mathcal{F}$ and $\mathcal{W}$, respectively. Unlike many-to-one matching, in this model each worker can work for many firms and each firm can hire a group of workers. Every worker w and every firm f has a quota, denoted q_w and q_f respectively, meaning that a worker w can work for at most q_w firms, and a firm f can hire at most q_f workers. Moreover, each worker has a preference list over subsets of firms and each firm has a preference list over subsets of workers. A similar set to the Ch_f set defined previously for firms is defined for workers, denoted $Ch_w(S)$, representing the worker w' most preferred subset of the set of firms S. Thus, the outcome of the matching is an assignment of a set of workers to each firm and a set of firms to each worker. Formally, a many-to-many matching μ is defined as follows:

Definition 3.11. *A matching μ is a mapping from the set $\mathcal{F} \cup \mathcal{W}$ into the set of all subsets of $\mathcal{F} \cup \mathcal{W}$ such that for every $w \in \mathcal{W}$ and $f \in \mathcal{F}$ [14]:*

1. *$\mu(f)$ is contained in $\mathcal{F}$ and $\mu(w)$ is contained in $\mathcal{F}$;*
2. *$|\mu(f)| \leq q_f$ for all f in $\mathcal{F}$:*
3. *$|\mu(w)| \leq q_w$ for all w in $\mathcal{W}$;*
4. *w is in $\mu(f)$ if and only if f is in $\mu(w)$.*

Now, we define a new type of preferences over sets of agents which is stronger than the two previous conditions defined in many-to-one section: substitutability and responsiveness.

Definition 3.12. *Separability means that for any agent i's set of potential partners S, $S\backslash b \prec_i S \cup b$ if and only if $b \prec_i \emptyset$.*

As for many-to-one matching, the instability in many-to-many market is not only caused by pairs of agents, since an agent of each side, might be matched to a group of players of the other side. Thus, the market instabilities might be caused by coalitions that consist of a group of workers and firms. In other words, a matching μ might be instable due to a set of firms and workers, that could arrange together and have more preferred partners than the assigned partners under the matching μ. Different stability concepts were introduced in the literature on many-to-many matchings. Hereafter, we define the most general ones.

Definition 3.13. *A matching μ is pairwise-stable if there no agents k and l who are not partners, but by becoming partners, possibly dissolving some of their partnerships given by μ to remain within their quotas and possibly keeping other ones, can both obtain a strictly preferred set of partners.*

Definition 3.14. *A matching μ is in the core (corewise-stable) if there is no subset of agents who by forming all their partnerships only among themselves, can all obtain a strictly preferred set of partners.*

Definition 3.15. *A matching μ is setwise stable if there is no subset of agents who by forming new partnerships only among themselves, possibly dissolving some partnerships of μ to remain within their quotas and possibly keeping other ones, can all obtain a strictly preferred set of partners.*

The authors in [23] proved that setwise stability is equivalent to the group stability introduced by [22] for the college admissions problem with responsive preferences. The author in [22] showed that setwise stability is the strongest stability concept by including the pairwise and corewise stability concepts. To prove it, the authors gave an example where there is a matching which is pairwise stable and corewise stable but is not setwise stable. The players' preferences in this example satisfy separability condition which is stronger than substitutability and responsiveness.

The main results on stability of many-to-many matching, given by the authors in [**?**], are as follows.

Theorem 3.6. *In the many-to-many matching model with responsive preferences, the set of pairwise stable matching is non-empty for all preferences, but a stable matching need not to be in the core.*

Theorem 3.7. *In many-to-many matching, when all the agents have substitutable preferences, the set of pairwise stable matchings is nonempty.*

These Theorems imply that even when a pair worker-firm cannot arrange together to have better partners, a larger coalition consisting of many workers and firms could give to all of them a more preferred assignments. Theorem 2.7 was proved by proposing the following algorithm which always find a pairwise stable matching.

- Step1: Each firm f makes offers to every worker w in the set $Q(f,1) = Ch_f(\mathcal{W})$;
- Step k:
  - (i) Each worker w rejects any offers received so far that are not in the set $Ch_w(O(w,k-1))$, where $O(w,k-1)$ is the set of offers w has received in steps 1,...$k-1$,
  - (ii) Each firm f who has received at least one rejection in the part (i) of step k and who now has (nonrejected) offers oUEstanding to the students in the set $N(f,k)$, and has been rejected in steps $1,...,k$ by the workers in the set $R(f,k)$, makes (or renews) offers to his most preferred set of workers $Q(f,k)$ in the collection of sets $\{Q \subset \mathcal{W} - R(f,k)$ such that Q contains $N(f,k)\}$. (Subtraction of sets is denoted -). That is, f's oUEstanding offers at the end of step k include all those issued at previous steps and not yet rejected and none of those that have already been rejected.
- The algorithm stops at any step$k = T$ at which no rejections are issued, and the resulting matching is μ such that $\mu(f) = N(f,T)$ for each f in $\mathcal{F}$.

Moreover, authors in [22] showed that there exists matching that every worker likes it at least as well as any other stable matching. Similarly, there exists a matching that every firm likes at least as well as any other matching.

Theorem 3.8. *When preferences are substitutable, the set of pairwise stable matchings is non-empty and there are firm-optimal and worker-optimal pairwise stable outcomes.*

There are many other stability and preferences concepts that were introduced for many-to-many matching such as strong separability and fix-point set [19]. While substitutability means that if a worker is chosen from a given set of workers, he is also chosen from a smaller set of workers; strong substitutability means that if a worker is chosen from a given set of workers, he is also chosen from a less preferred set of workers. Strong substitutability is stronger than substitutability but is weaker than separability and not stronger than responsiveness. The formal definition is given as follows:

Definition 3.16. *strong substitutability requires: if hiring w is optimal when the set of available workers is $\{w\} \cup S'$, and the firm prefers S' to S, then hiring w must still optimal when the set of available workers is $\{w\} \cup S$.*

The main result including strong substitutability is the following:

Theorem 3.9. *If firms' preferences are substitutable and workers' preferences are strongly substitutable, the theory of many-to-many matching parallels the theory of many-to-one matchings: the setwise stable set equals the pairwise stable set.*

The fix-point set is defined as follows:

Definition 3.17. *fix-point set is a matching where each agent k is choosing his/her best set of partners, out of the set of potential partners who, given their current match, are willing to link to k.*

The main results on the fix-point set are given in [19]:

Theorem 3.10. *If preferences are substitutable, the fix-point set is not empty. And the fix-point set equals the set of individually rational and pairwise matchings*

The authors gave also an algorithm that find a fix-point matching and proved that if firms' preferences are substitutable and workers' preferences are strongly substitutable, the fix-point set equals the set of setwise stable matchings, and a matching in the fix-point must be in the individually rational core. Thus, setwise stable matchings exist, the individually rational core is nonempty, and they gave an algorithm that finds a matching in the individually core that is setwise stable. The same results hold when firms' preferences are strongly substitutable and workers' preferences are substitutable [19].

3.5 Classification of Matching Games

Here, we give a classification of matching games taking into account the parameters that could be considered when defining players' preferences. It should be noted that the matchings in each of these classes might be a one-to-one, many-to-one or many-to-many matchings. The canonical games class is the most popular matching class and it represents the basic model without taking into account any additional settings. Matching with externalities class and matching with dynamics class consider the external effects and the environment changes over the time, respectively. Here, we give a brief overview on these three classes of matching games:

- **Canonical Matching Games:**
  Canonical matching represents the basic matching class, in which any player n preference function depends only on the player n himself and the potential partner k in the opposite side. This class could be mainly used as a basic model to develop more sophisticated matching classes. In the context of telecommunication networks, this class can be used to model a basic scenario for proactive resource allocation.

- **Matching with Externalities:**
  In the studied matching problems (marriage market, college admissions and labor markets) agents have preferences only over the opposite set of agents. However, in many situations agents might define their preferences according the others matched agents. For instance, in the marriage market jealousy may play a main role, since a man or a woman might decide do not match at all or to accept any proposal if two specific agents are matched to each other. This kind of effects are called

externalities. The other considered feature in this class is called *peer effect*, which corresponds to the mutual effects of players that are assigned to the same player in the opposite set. When the externalities are present, the players should consider how other players would react to the deviation. This class of games is mainly used when the interferences are considered in telecommunication networks, and control external factors such as the cell radius by the SBSs.

- **Matching with Dynamics:** In general, the environment in which players interact is not static and many changes may occur. For instance, if we consider the labor market of firms and workers, the environment changes could be the set of firms that can at any time period dismiss their current employees or hire new ones. These firms did not commit themselves to their employees and are called *active firms*. On the other hand, workers are said *active workers* when they also have no commit and can leave their current employing firm if they have a better proposal from another firm. In the context of wireless communication, the environment changes might be for instance, the mobility of users or the backhaul state according to which the players could be better off under another matching. Thus at the beginning of each time period, based on the constraints and the previous matching, a new matching is defined between players.

3.6 Conclusion

Matching games was first introduced to explain and predict the behavior of complex economic systems. In this chapter we focused on the two-sided matching markets, where the real-world system consists of two sets of players and each player belongs to only one side. The main goal in the system is to assign to each player one or many players of the other side based on their preferences and their quotas which represent respectively, the most preferred and the number of players they want to be matched to. Matching theory represents one of the most important game theoretic approaches that were extended and applied to telecommunication systems, mainly for resources management. In fact, network's resources such as backhauling and spectrum are limited whereas the mobile data traffic has been grown exponentially. In order to design efficient techniques for resources management, the resources allocation problems are formulated as optimization problems. However, most of the formulated problems are combinatorial with exponential complexity which prompted the integration of game theoretic approaches to solve these problems.

The major result on two-sided matching games was given by Gale and Shapley in their prominent paper [12]. The authors studied the marriage market where the goal is to match each man of the market to a woman. Their main result is the deferred acceptance algorithm they proposed. The algorithm produces a stable matching between the two sides of the market, in the sense that no agent will have the incentive to leave his partner for another more preferred player. Thus, the essential coalitions that might cause instabilities in this market are pairs of agents. This concept of stability is known as pairwise stability. The second most popular matching problem is the college admissions market where a set of students is assigned to each college. This model was proved to be equivalent to the marriage model and hence, the deferred acceptance algorithm provide also a pairwise stable matching for this market. However, in this model, the coalitions that cause instabilities might consist of more than a pair of players. Thus, the setwise stability concept is considered as being more appropriate for multiple partners models. In [22], the authors proved that under certain restrictions on the preference lists of players, the pairwise and setwise stability concepts are equivalent and the deferred acceptance algorithm provides a non-empty setwise stable matching. The more general matching model is the firms and workers labor market in which a worker can work for a group of firms and each firm can hire a group of worker. This model is not more complicated than the two previous models and the results on college admissions problem could not being extended. The main results on this model were given by [18] giving an algorithm that produce a pairwise stable matching when the agent's preferences are substitutable.

In the next chapter, we show how the caching problem in SCNs can be modeled as a matching game. Principally, we model the caching problem as a one-to-one matching game while considering the users expectation in the players' preference functions.

4 Proposed Approach

4.1 Introduction

The limited backhaul capacity is considered as one of the most important issues in SCNs. In fact, the success and popularity of Youtube, NetFlix and other video content providers have mainly contributed in the growth of network traffic. Moreover, the high bitrate of live streaming and Video on Demand (VoD), compared to Web pages and other content, generates most of the traffic and produces network congestion. This increases both delays and interruptions in video streaming which badly affect users' QoE resulting in churn users and inducing lost revenue for content providers. In this chapter we propose a caching strategy for content providers that aims to ensure a fast response time and reduce the load of backhaul links.

The main work [8] that introduced caching strategy to deal with the limited backhaul capacity formulated the caching problem as an optimization problem, in which the goal is to reduce the download time of content in the network. While most of the resource allocation problems in SCNs are formulated as optimization problems, our solution is based on matching theory which provides practical solutions for hard optimization problems. We model the storage problem as an iterative one-to-one matching between two opposite sets of CPSs and SBSs, respectively. On one hand, content providers has sets files stored in their own data servers and they aim to cache the files closer to the users in SBSs, in order to reduce the delivery time without lags and interruptions. On the other hand, SBSs have a limited storage capacity and aim to store the most popular files in order to reduce the backhaul load in the network.

The authors in [8] proposed to start by caching the most popular files, i.e., the files that have a hight probability to be requested by the users. However, depending on the preferences of users that are connected to each SBS over time, the popularity of files differs from a SBS to another SBS. In our work, instead of considering the global popularity of files, we define a local popularity at each SBS according to which SBSs choose the files they would store firstly. This parameter represents the preference relation of each SBS over the files stored in the CPSs. The preference relation of each CPS is defined depending on the delivery time at each SBS. In other words, a CPS would prefer to store its files in a SBS that offer the minimal download time to the users.

This chapter is organized as follows. In section 2, we describe the model and formulate the optimization problem for caching in SCNs. In section 3, we model the caching problem as an iterative one-to-one matching game and define the preferences of each team. In section 4, we propose an iterative algorithm which find a stable and optimal matching between CPSs and SBSs.

4.2 System Model

We consider a network in which content providers store their files in their own servers and would like to cache the files deeper in the network in order to ensure a better QoE for users. We suppose a network comprising N users equipments (UEs) denoted $\mathcal{N} = \{u_1, u_2, \ldots, u_N\}$, served by K CPSs denoted $\mathcal{C} = \{c_1, c_2, ..., c_K\}$ and M SBSs denoted $\mathcal{M} = \{s_1, s_2, \ldots, s_M\}$ (see Figure 2). Each CPS c_i is connected to the SBSs through low-rate backhaul links of capacities $B_i = [b_{i1}, b_{i2}, ..., b_{iM}]$ through which SBSs download the files from CPSs. The capacities of all the backhaul links that connect the CPSs to the SBSs are summarized in the matrix B:

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_k \end{bmatrix} = \begin{bmatrix} b_{11} & ... & b_{1M} \\ b_{21} & ... & b_{2M} \\ \vdots & \ddots & \vdots \\ b_{K1} & ... & b_{KM} \end{bmatrix} \in \{0, \mathbb{Z}^+\}^{K \times M}, \tag{1}$$

where b_{ij} represents the capacity of the backhaul link connecting the i^{th} CPS to the j^{th} SBS.
The downloaded files are then cached in storage units of high but limited storage capacities $Q = [q_1, q_2, ..., q_M]$. The storage capacity is defined by the number of files that each SBS can store. Thus, each SBS can locally serve its UEs over the radio links, with different rates given in the matrix W.

$$W = \begin{bmatrix} w_{11} & ... & w_{1N} \\ w_{21} & ... & w_{2N} \\ \vdots & \ddots & \vdots \\ w_{M1} & ... & w_{MN} \end{bmatrix} \in \{0, \mathbb{Z}^+\}^{M \times N}, \tag{2}$$

where w_{ij} represents the capacity of the radio link connecting the i^{th} SC to the j^{th} UT.

In this scenario, we aim to produce a proactive download of files which we define as follows:

Definition 4.1. *Proactive Caching: A caching is said proactive if SBSs can predict the users' requests and download ahead of time the related files.*

In the *reactive caching* model, files are downloaded when the users request them, thus the following requests for the same files can be served locally. However, due to the limited storage capacity of the SBSs and the hight number of requests, most of the requests cannot take benefit from the caching strategy. For instance, if a file f is cached when a user requests it, this file might not be requested subsequently. Moreover, during peak hours, the load of the backhaul is very high and all the users need to be served at the same time resulting in a very large delivery time for users.

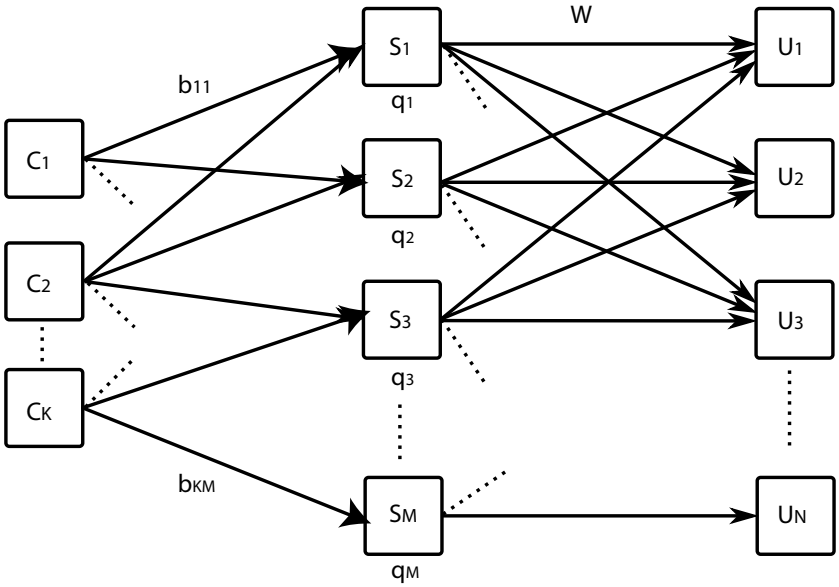

Figure 2: Network Model.

Let's consider that the users might request files among a library of F files $\mathcal{F} = \{f_1, f_2, \ldots, f_F\}$ and each file is owned by a specific content provider and stored in its corresponding CPS. To decide which files should be cached, the SBSs predict the UEs' requests over a time period $D = t_0..t_d$. Each SBS captures the user n's request of the file f at time t by a variable $\mathbb{1}_{n,t}(f)$ defined as follows:

$$\mathbb{1}_{n,t}(f) = \begin{cases} 1 & \text{if UE } n \text{ requests file } f \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

For ease of analysis, in our formulation we make the following assumptions:

1. We assume that all the files have the same size $L_i = L \; \forall i \in 1..F$. This assumption can, for example, reflect the practical case in which files are broken into packets of the same size.

2. Each SBS can either store a file completely or do not store it at all. In other words, the SBSs do not store partial files.

3. All the SBSs connected to a given UE, predict the same requests coming from this UE.

4. Each file is owned by only one content provider and thus stored in the its own CPS.

Now, suppose at the beginning of the time window D, each SBS sends its predictions to a central application server. Thus, the central application will be aware of all the R requests. The total number of requests R is defined as follows:

$$R = \sum_{t=t_0}^{t_d} \sum_{n=1}^{N} \sum_{f=1}^{F} \mathbb{1}_{n,t}(f). \tag{4}$$

Thus, according to this information, the central application server would define a proactive caching strategy such as the CPSs deliver the requested files to the SBSs which in turn serve their UEs locally reducing the backhaul load. Let us define the two vectors $\mathbf{t} = [\hat{t}_1, \ldots, \hat{t}_R]'$ and $\mathbf{x} = [\hat{c}_1, \ldots, \hat{c}_R]'$ as the starting delivery time and the caching SC of the requested files, respectively (where, $\mathbf{t}'$ is the transpose of the row $\mathbf{t}$). We formulate the optimization problem as a maximization of the satisfied requests in time, i.e., cached before their arrival time, under the capacity constraints:

$$\begin{aligned} \underset{\mathbf{t},\mathbf{x}}{\text{maximize}} \quad & \frac{\sum_{r_i \in \mathbf{r}} \mathbb{1}\left\{t_w^{r_i} + t_s^{r_i} \leq t^{r_i}\right\}}{R} \\ \text{subject to} \quad & B \leq B^{max}, \\ & Q \leq Q^{max}, \\ & W \leq W^{max}, \end{aligned} \tag{5}$$

where $t_w^{r_i}$ is the waiting time of a request at the CPS to be served, and $t_s^{r_i}$ is the required time for a CPS to send the requested file. B^{max}, Q^{max} and W^{max} represent the capacity constraints of the backhaul links, storage units and radio links, respectively.

The formulated optimization problem is a combinatorial decision problem which depends on the number of SBSs and UEs in the network as well as the length of the time period D. To solve this problem, we formulate it in the next section, as a one-to-one matching game and we propose an iterative deferred acceptance algorithm as a solution.

4.3 Problem Formulation

To model the system as a matching game, we consider the two sets $\mathcal{C}$ of CPSs and $\mathcal{M}$ of SBSs as two teams of players (agents). In this model each content provider wants to store its files in the SBSs and each SBS wants to store files in the limit of its storage capacity. Thus, the goal is to match files to the appropriate SBSs in order to reduce the load of backhaul links and improve the users' QoE. It should be noted that the files are matched to the SBSs but in our model, CPSs acts on behalf of the files and each of them decides on its own files. Moreover, all the files that belong the one content provider, i.e. the files that are stored at a specif CPS, are not necessarily cached at the same SBS. For ease of analysis, we consider that each content provider aims to cache one file among the files stored in its CPS. This assumption could be considered as a special case of the model in which the most popular file is firstly cached by the CPS. The library of files $\mathcal{F}$ is scattered among the CPSs such us each content provider store his files in his CPS. Based on the preferences of the CPSs and the SBSs, the central server application decides in which SBSs to cache each file.

4.3.1 Preferences of the Content Provider Servers

The library of files $\mathcal{F}$ is scattered among the CPSs such as each content provider stores his files in his CPS. The goal of content providers is to cache their files such as the UEs have a hight QoE which allows them to keep their subscribers for a long time period. In our model we mainly take into account the delivery time of the UEs. In fact, a CPS would prefer to store the files in the SBSs that offer a smallest download time for the UEs. Let us start by defining the required time for the n^{th} UE to download a file f of size L from the m^{th} SBS, knowing that the file f is stored in the k^{th} CPS:

$$T_D = \frac{L}{\min(b_{km}, w_{mn})}. \tag{6}$$

In fact, the download time depends on the capacity of the backhaul and the radio links. Since the file is first downloaded by the SBSs which then serve the UEs, the download time for a UE equals to the maximal required time among the needed times to download the file to the SBS and to download it to the UE.

When many UEs request the same file from a SBS, the expected download time is computed as follows:

$$\overline{T_D} = \frac{L}{\min(b_{km}, \frac{\sum_{n=1}^{N} w_{mn}}{N})}. \tag{7}$$

Thus, the CPSs can define their preferences list over the set of SBSs based on the delivery time. In other words, the most preferred SBS for a CPS to cache a file is the one that offers the smallest delivery time for its subscribers (UEs).

4.3.2 Preferences of the Small Base Stations

The objective of SBSs is to reduce the backhaul links load by storing the files before the users requests these files. This process is beneficial when SBSs download the files during off-hours (e.g., during the night) and serve the users locally during pick hours (e.g. during the day). Thus, a SBS would prefer to store first a file that it has the highest local popularity, i.e., the most requested file at that SBS.

Now, we define the local popularity of file f_i at the m^{th} SBS:

$$LP_{s_m}^{f_i} = \frac{\sum_{n=1}^{N} \sum_{t=t_0}^{t_d} \mathbb{1}_{n,t}^{s_m}(f_i)}{\sum_{j=1}^{F} \sum_{n=1}^{N} \sum_{t=t_0}^{t_d} \mathbb{1}_{n,t}^{s_m}(f_j)}. \tag{8}$$

4.4 Matching Algorithms

In this section, we propose an extension of the deferred acceptance algorithm for the marriage problem and apply it to the caching problem.

4.4.1 Conventional Deferred Acceptance Algorithm

First, let us describe the deferred acceptance algorithm to define a one-to-one matching between files and SBSs and then, we propose an iterative version of this algorithm to solve the global matching problem. The conventional deferred acceptance works as follows:

- Step 1: Each CPS sends a request to its most preferred SBS, i.e., the SBS offering the smallest delivery time for the related file.
- Step 2: Each SBS_i accepts to store the most preferred file and reject the others.
- Step 3: CPSs that were rejected at step 2 propose to their second choices.
- Step 4: Each SBS chooses its favorite CPS (i.e., file) from the new proposing SBSs and the last chosen one, and rejects the others.
- Step 5: Redo Step 3 and Step 4 until each CPS is matched or rejected by all the SBSs.

4.4.2 Iterative Deferred Acceptance Algorithm

The conventional deferred acceptance algorithm cannot be applied to our problem as proposed, since one file could be assigned to several SBSs. therefore, we generalize the previous algorithm to the case where SBSs can store more than one file. The proposed algorithm works as follows:

- Step 1: Process Steps 1-5 above of the deferred acceptance algorithm.
- Step 2: update the preferences of CPSs and SBSs according to the previous matching by subtracting the satisfied UEs and decreasing the storage capacity of SBSs by the number of accepted files (one or zero).
- Step 3: Redo the Step 1 and Step 2 until the storage capacities of all SBSs reach zero.

4.5 Stability and Optimality of the Proposed Algorithm

Here, we study the characteristics of the proposed deferred acceptance algorithm. As we showed in Chapter 3, stability and optimality are the most important features of the matching algorithms. In our context, stability means there does not exist a pair of SBS and CPS that prefer each other to the partner they receive by the iterative matching algorithm.

Theorem 4.1. *The iterative deferred acceptance algorithm is stable and optimal.*

Proof. The proof of the stability and optimality of the proposed iterative deferred acceptance algorithm can be considered as an extension of the Theorem 3.1 and Theorem 3.2. In fact, one iteration of the algorithm corresponds to the marriage problem for

which the deferred acceptance algorithm finds a stable matching and provides an optimal matching. Thus, the iteration of the deferred acceptance algorithm is also stable and optimal. □

4.6 Conclusion

In this chapter, we first formulated the caching problem as an optimization problem that aims to maximize the number of satisfied requests proactively, i.e., the files are cached before the users request them. However, this formulated problem is a combinatorial problem which depends on the number of SBSs in the network. To solve the caching problem, instead of using the conventional solutions such as heuristic and meta-heuristic algorithms that do not ensure an optimal solution, we explored one of the game theoretic tools that offers practical solutions for such problems. In fact, we modeled to caching problem as an iterative one-to-one matching game between CPSs and SBSs. On one hand, the CPSs aim to reduce the delivery time of the UEs in order to satisfy and keep their subscribers for a long period. Thus, the CPSs define their preferences over the set of SBSs based on the delivery time offered by each SBS. On the other hand, SBSs aim to reduce the backhaul load by storing the most popular files locally. While in the previous works [8], the authors exploited the global popularity of files which corresponds the request rate of each file over all the SBSs, we restricted the files' popularity to each SBS. This choice is motivated by the fact that even if a file is globally popular, depending on the SBSs' location and the preferences of users that might connect to them, the files' popularity changes from a SBS to another SBS. For instance, a file can be requested by all the users connected to a given SBS while that same file is not requested at all by the users connected to another SBS. So even even if the file has the highest global popularity it is not efficient to store it in the some SBSs.

After the definition of the players' preferences, we proposed an extension of the Gale and Shapley algorithm. The goal behind this extension is to exploit the fact that each SBS might store more than one file. One could think that since each SBS can store many files and each file might be stored in many SBSs, why do not we model the caching problem as a many-to-many matching game. The main reason for which we did not choose the many-to-many formulation is because in our scenario many SBSs might predict the same UE's request but the UE is served by only one SBS. Thus, if all the SBSs store the same file without taking into account the neighboring SBSs, the process will induce a waste of storage space. To take into account this parameters more elaborated preferences functions should be defined, as we can also use a matching game with externalities to take into account the peer effects.

In following section we provide the numerical results to show the performances of the proposed algorithm.

5 Experimental Results

To study the performances of the proposed caching algorithm, we implemented our scenario and applied two algorithms. The first caching algorithm is the iterative deferred acceptance algorithm we proposed In Chapter 3, and the second one is a baseline algorithm in which files are stored randomly in the SBSs. We performed the simulations and recorded the numerical results for different configurations of the SBSs' storage capacity. For this purpose, we define a ratio γ that represents the proportion of available storage space:

$$\gamma = \frac{Q^{\max}}{M.F.L}. \tag{9}$$

We performed the simulations for five different storage ratio $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$. Where $\gamma = 0$ corresponds to the case in which all the SBSs are saturated or are not equipped with storage units, i.e., the quota of all the SBSs $q_i = 0$, $\forall i$. $\gamma = 1$ corresponds to the case in which all the files can be cached in the storage units, i.e., the quota of all the SBSs $q_i = F$, $\forall i$. Table 1 summarizes all the parameters that we used in our simulations.

Table 1: Simulation parameters.

Parameter	**Values**	**Description**
D	2000	time window
M	6	number of SBSs
K	6	number of CPSs
N	16	number of UEs
B^{max}	20	total capacity of backhaul links (Mbit)
W^{max}	150	total capacity of wireless links (Mbit)
F	128	number of files
l_i, $\forall i$	256	length of files (Mbit)
b_{ij}, $\forall i, j$	5	backhaul link capacities (Mbit)

The simulations using both of the algorithms were repeated 50 times and averaged. The impact of the two algorithms on the backhaul usage was analyzed depending on the total number of requests R in the network. The obtained numerical results for the simulations of the matching-based algorithm and random caching are shown in Figure 3 and Figure 4, respectively.

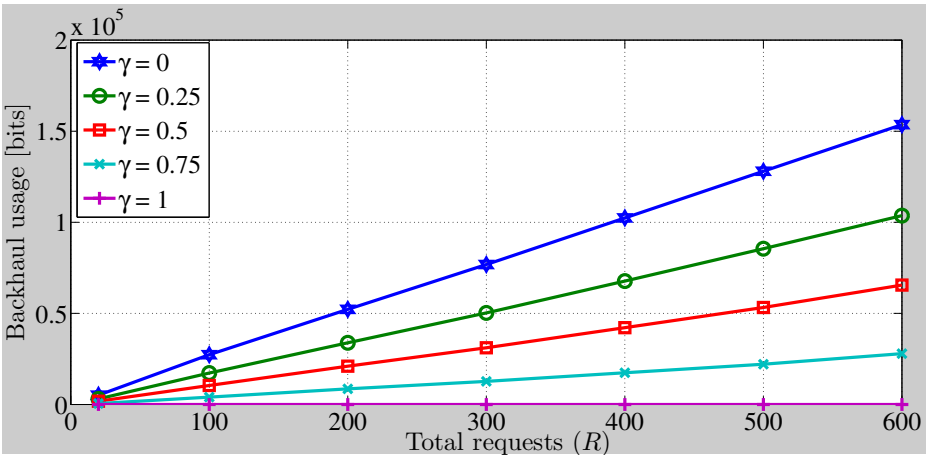

Figure 3: The backhaul usage using Iterative Deferred Acceptance Algorithm.

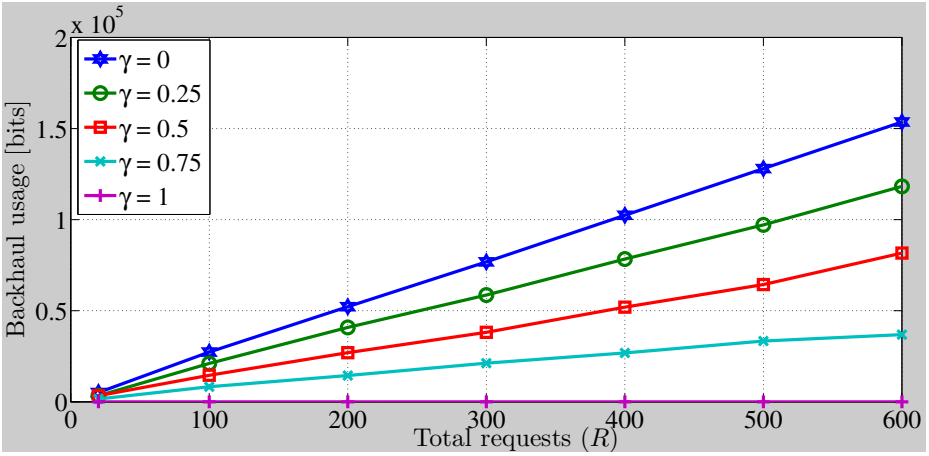

Figure 4: The backhaul usage using Random Caching Algorithm.

In both of the Figures, we notice that when there is no storage space to cache the files ($\gamma = 0$), the backhaul link usage is higher than the other configurations. Oppositely, when the storage units have the capacity to store all the files ($\gamma = 1$), the backhaul usage equals 0. However, we can see the difference between the two approaches for the other values of γ ($\gamma \in \{0.25, 0.5, 0.75\}$). In fact, we notice that the caching algorithm based on the deferred acceptance algorithm reduce the backhaul usage by 10-15 %

6 Conclusion

In this thesis, we mainly addressed the backhaul issue in SCNs. We started our work by presenting an overview on SCNs, the architecture of these networks, their advantages, and their limitations. Then, we discussed the novel promising works that deal with the backhaul issue through caching strategies. We noticed that most of these works are based on classical optimization tools that mostly do not find the optimal solutions or often the proposed algorithms have a high complexity and could not be applied in real systems. In the second chapter, we explored game theory that provides mathematical tools to model hard problems and find practical solutions. In particular, we introduced matching theory and presented the main results in economic literature. Finally, we formulated the storage problem in SCNs as an iterative one-to-one matching and provided an extension of the deferred acceptance algorithm and showed that it always provide a stable and optimal matching game between files owned by content providers and SBSs. The experimental results show that the proposed algorithm can reduce the backhaul load by up to 15 % compared to a random caching policy.

From a scientific point of view, this internship allowed me to have a strong introduction to game theory an to start working on a paper under the title "Many-to-Many Matching Games for Proactive caching in Small Cell Networks". In that paper we exploit novel, under-explored information, extracted from practical network dimensions such as smartphones, social networks, or geo-location information so as to proactively cache data. We modeled a caching problem as a many-to-many matching which is very new approach and still not well understood. To our knowledge only two works in literature have exploited this class for resource allocation in wireless networks. For sack of time we could not finish and present that work.

From a future Ph.D student point of view, this internship has been very rich. First, it enabled me to acquire knowledge on game theory in general and matching theory in particular, which is one of the aspect of my Ph.D subject: game theory applied to the wireless networks. Secondly, it was a good opportunity to interact and work with the dynamic team members of the Alcatel Lucent Chair on flexible Radio. I would like to thank especially, my internship director Prof. Mérouane Debbah and Dr. Walid Saad, for offering me this opportunity and accepting to advise me.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016," *White paper*, `http://www.cisco.com/en/US/netsol/ns827/networking_solutions_white_papers_list.html`, May.2012.

[2] T. Q. S. Quek, G. D. L. Roche, I. Guven, "Small Cell Networks", Cambridge University Press, Edition 1, Jun. 2013.

[3] Small Cell Forum, "Small cells - what's the big idea?", Feb. 2012.

[4] J. Hoydis, M. Kobayashi, M. Debbah "Green Small-Cell Networks," *IEEE Vehicular Technology Magazine*, vol. 6, no. 1, pp. 37-43, Mar. 2011.

[5] Y. Bouguen, E. Hardouin, F. -X. Wolff , "LTE et réseaux 4G," *Edition Eyrolles*, Oct. 2012.

[6] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, G. Caire, "FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers," *Proceedings of the IEEE Infocom*, pp. 1107-1115, Mar. 2012.

[7] N. Golrezaei, A. G. Dimakis, A. F. Molisch, "Wireless Device-to-Device Communications with Distributed Caching," *IEEE International Symposium on Information Theory Proceedings*, pp. 2781-2685, Jul. 2012.

[8] S. Boyd, L. Vandenberghe, "Convex Optimization," `http://www.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`, Stanford University, 2004.

[9] S. Fujishige, "Submodular Functions and Optimization," Second Edition, Elsevier, 2005.

[10] V. Etter, M. Kafsi, E. Kazemi, "Been there, Done That: What Your Mobility Traces Reveal about Your Behavior", *Nokia Mobile Data Challenge- Next Place Prediction*, Jun. 2012.

[11] C- M. Huang, J. J- C. Ying, V. S. Tseng,"Mining Users' Behaviors and Environments for Sema,tic Place Prediction", *Nokia Mobile Data Challenge*, Jun. 2012.

[12] D. Gale, L. Shapley, "College Admissions and the Stability of Marriage," *American Mathematical Monthly*, pp. 9-15, 1969.

[13] A. E. Roth, M. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis,* Econometric Society Monograph Series, Cambridge University Press, 1990.

[14] A. E. Roth, "A Natural Experiment in the Organization of Entry Level Labor Markets: Regional Markets for New Physicians and Surgeons in the U.K," *American Economic Review*, vol. 81, pp. 415-425, 1991.

[15] A. E. Roth, "Deferred Acceptance Algorithm: History, Theory, Practice, and Open Questions," *Int. J. Game Theory*, vol. 36, pp. 536-569, 2008.

[16] D. Gale, M. Sotomayor, "Some Remarks on the Stable Matching Problem," *mimeo*, 1983.

[17] A. E. Roth,"The College Admissions Problem is not Equivalent to the Marriage Problem," *Journal of Economic Theory*, vol. 54, pp.425-427, 1985.

[18] M. Sotomayor, "Three Remarks on the Many-to-Many Stable Matching Problem," *Mathematical Social Sciences*, vol. 38, pp. 55-70, 1999.

[19] F. Echenique, J. Oviedo, "A Theory of Stability in Many-to-Many Matching Markets," *Theoretical Economics*, vol.1, pp. 233-273, 2006.

[20] C. Blair, "The Lattice Structure of the Set of Stable Matchings with Multiple Partners," *Mathematics of Operations Research*, vol. 13, pp. 619-628, Nov. 1988.

[21] I. Brito, P. Mesequer, "Distributed Stable Matching Problems," *Proceedings of Constraints Programming*, vol. 53, pp. 152-166, 2005.

[22] A. Roth, "Stability and Polarization of Interests in Job Matching," *Econometrica*, vol. 52, pp. 47-57, Jan. 1984.

[23] H. Konishi, M. Utku Unver, "Credible group-stability in many-to-many matching problems", *Forthcoming, Journal of Economic Theory*, Boston College, 2005.